Multi-GPU FFT Performance on Different Hardware Configurations

Kevin Roe
Maui High Performance Computing Center

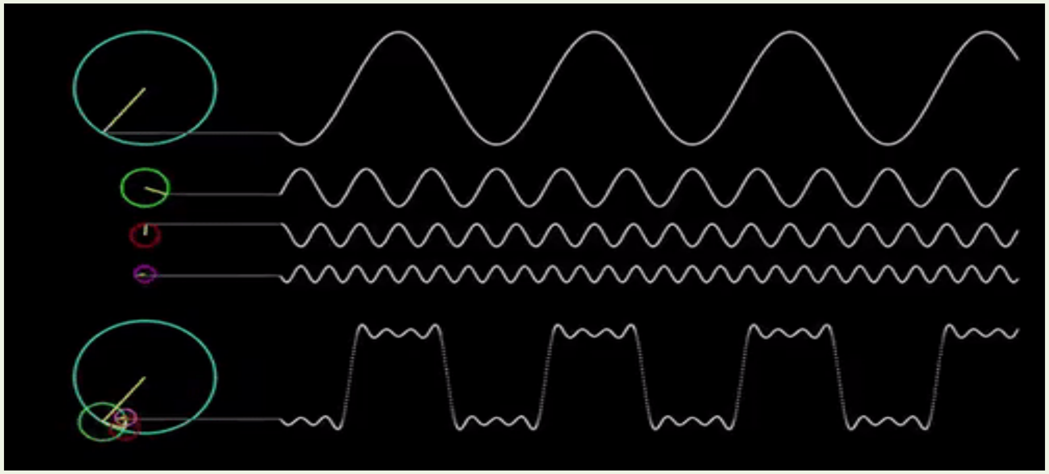
Ken Hester
Nvidia

Raphael Pascual
Pacific Defense Solutions

Advance Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center, March 19, 2019

Distribution A: This is approved for public release; distribution is unlimited

Fast Fourier Transform (FFT)

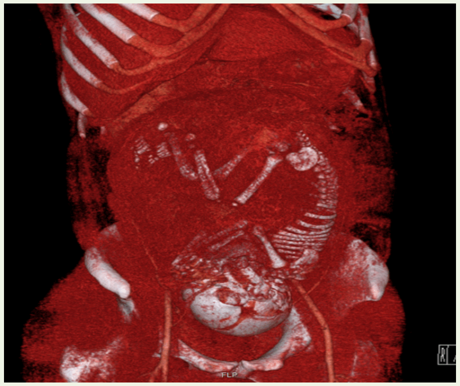


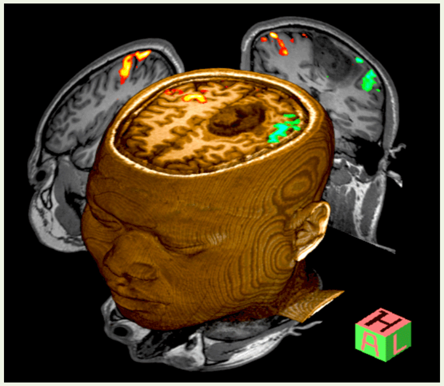
- **The Fourier transform**
  - Decomposes a function of time into the frequencies that make it up
  - Discretize then compute using FFTs
- **Motivating FFT based applications**
  - Digital Signal Processing (DSP)
    - Medical Imaging
    - Image Recovery
  - Computational Fluid Dynamics
  - Can require large datasets
- **Focus on large FFT problems that can not be solved in batch mode**
  - Benchmarking multi-GPU FFTs within a single node
    - CUDA functions
  - Collective communications
  - **Bandwidth and latency** will be strong factors in determining performance

Distribution A: This is approved for public release; distribution is unlimited

Medical Imaging

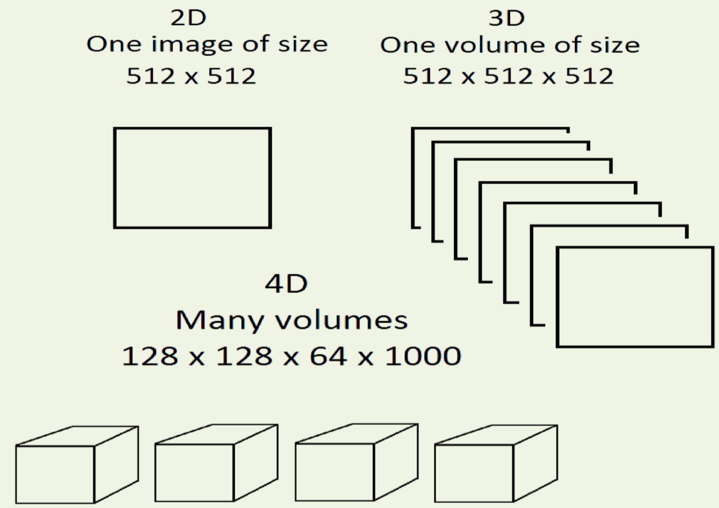
- **Correct high resolution imaging can prevent a misdiagnosis**
- **Ultrasonic Imaging**
  - Creates an image by firing & receiving ultrasonic pulses into an object
  - Preferred technique for real-time imaging and quantification of blood flow
    - Provides excellent temporal and spatial resolution
    - Relatively inexpensive, safe, and applied at patient's bedside
    - Low frame rate
  - Traditional techniques do not use FFT for image formation
  - Pulse plane-wave imaging (PPI)
    - Utilizes FFTs for image formation
    - Improved sensitivity and can achieve much higher frame rates
- **Computed Tomography (CT)**
  - Removes interfering objects from view using Fourier reconstruction
- **Magnetic Resonance Imaging (MRI)**
  - Based on the principles of CT
  - Creates images from proton density, Hydrogen (^1H)
  - Image reconstruction by an iterative non-linear inverse technique (NLINV)
    - Relies heavily on FFTs
  - Real-time MRIs require fast image reconstruction and hence powerful computational resources

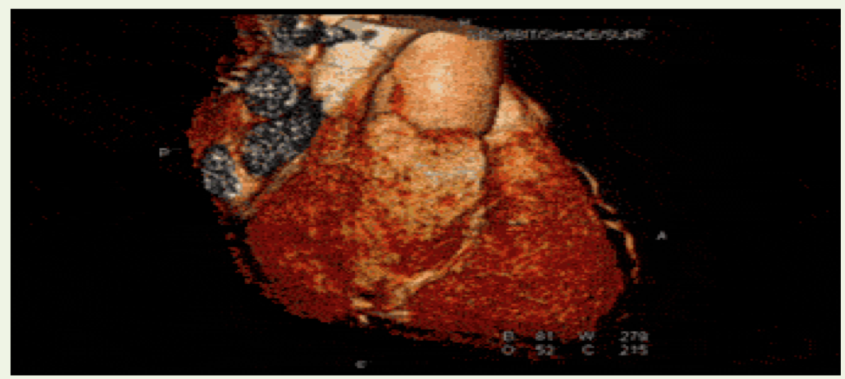




Medical Imaging (continued)

- **Multi-Dimensional requirements**
  - 2D, 3D, and 4D imaging
  - Traditional CT & MRI scans produce 2D images
  - Static 3D Volume (brain, various organs, etc.)
    - Combining multiple 2D scans
  - Moving objects incorporate time
    - 3D video image: multiple 2D images over time
    - 4D video volume: multiple 3D volumes over time
- **Supplementary techniques also require FFTs**
  - Filtering operations
  - Image reconstruction
  - Image analysis
    - Convolution
    - Deconvolution



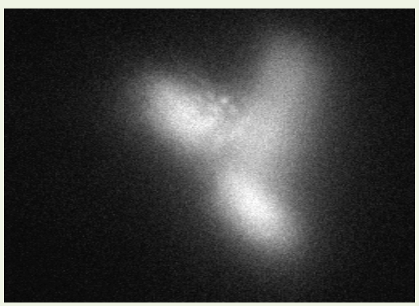


Distribution A: This is approved for public release; distribution is unlimited

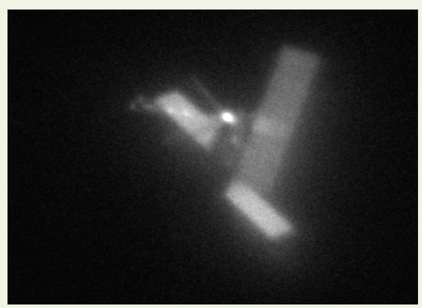
Image Recovery

- **Ground based telescopes require enhanced imaging techniques to compensate for atmospheric turbulence**
  - Adaptive Optics (AO) can reduce the effect of incoming wavefront distortions by deforming a mirror in order to compensate in real time
    - AO cannot completely remove the effects of atmospheric turbulence
  - Multi-frame Blind Deconvolution (MFBD) is a family of “speckle imaging” techniques for removing atmospheric blur from an ensemble of images
    - Linear forward model: $\boldsymbol{d_m(x) = o(x) * p_m(x) + \sigma_m(x)}$
      - Each of ***m*** observed data frames of the image data ($\boldsymbol{d_m(x)}$) is represented as a pristine image ($\boldsymbol{o(x)}$) convolved with a Point Spread Function ($\boldsymbol{p_m(x)}$) as well as an additive noise term ($\boldsymbol{\sigma_m(x)}$) that varies per image.
      - Ill-posed inverse problem solved with max likelihood techniques and is very computationally intense
    - Requires FFTs in its iterative process to calculate the object, producing a “crisper” image

Seasat



AO



MFBD

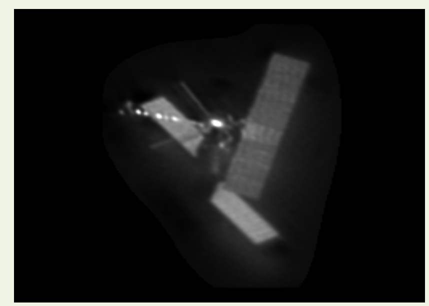
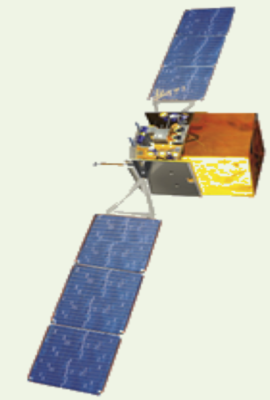
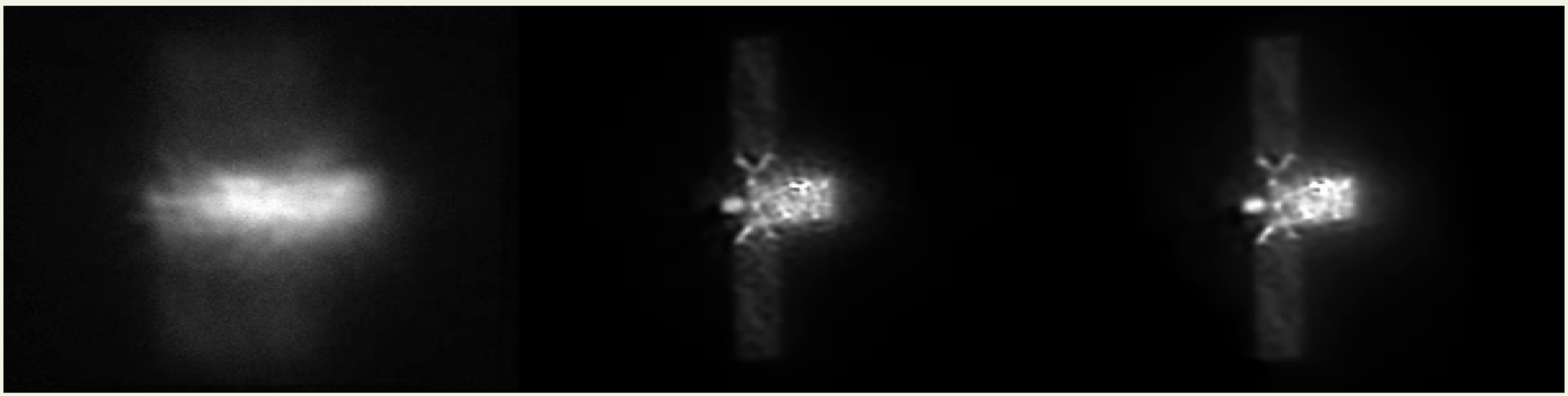


Image Recovery (continued)

- **Physically Constrained Image Deconvolution (PCID)**
  - A highly effective MFBD has been parallelized to produce restorations quickly
  - A GPU version of the code is in development
- **Fermi Gamma-ray Space Telescope: NASA satellite (2008)**
  - Study astrophysical and cosmological phenomena
  - Galactic, pulsar, other high-energy sources, and dark matter

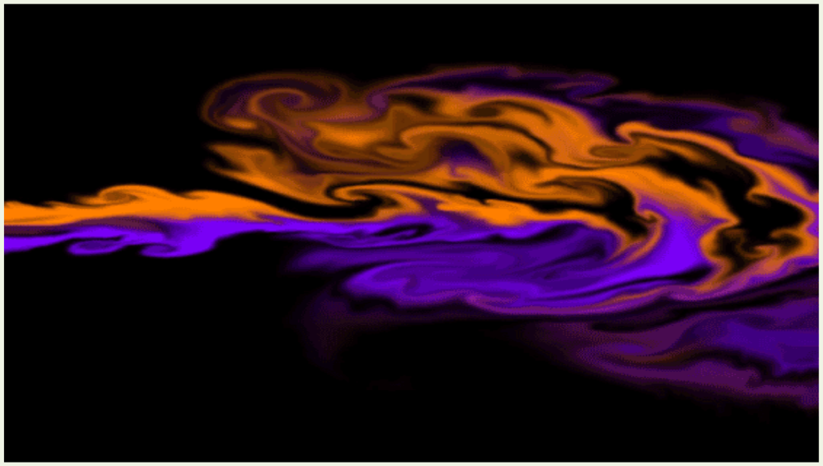


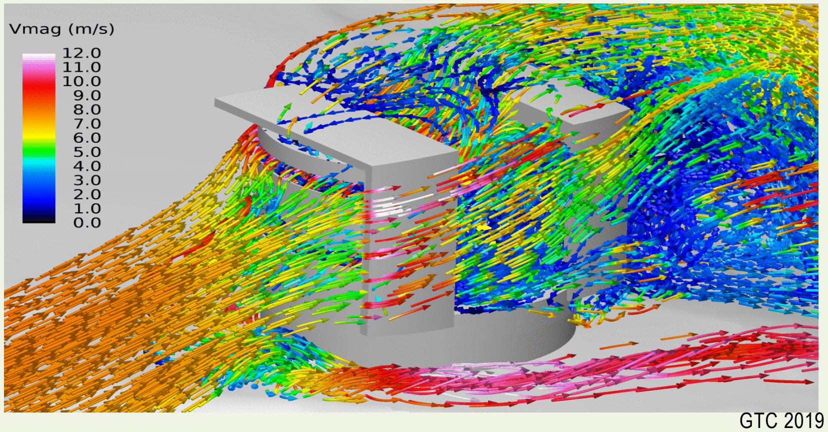


Distribution A: This is approved for public release; distribution is unlimited

Computational Fluid Dynamics

- **Direct Numerical Simulation (DNS)**
  - Finite Difference, Finite Element, & Finite Volume methods
  - **Pseudo Spectral method**: effectively solving in spectral space using FFTs
- **Simulating high resolution turbulence**
  - Requires large computational resources
  - Large % of time spent on forward and inverse Fourier transforms
  - Effective performance can be small due to its extensive communication costs
  - Performance would be improved with higher bandwidth and lower latency
- **Code examples that utilize FFTs on GPUs**
  - NASA's FUN3D
  - Tarang
  - UltraFluidX





Distribution A: This is approved for public release; distribution is unlimited

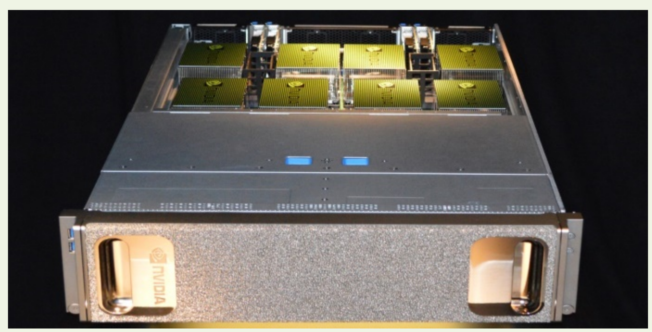
Benchmarking Multi-GPU FFTs

- **Represent large 3D FFTs problems that cannot fit on a single GPU**
  - Single precision Complex to Complex (C2C) in-place transformations
    - C2C considered more performant than the Real to Complex (R2C) transform
    - In-place – reduces memory footprint and requires less bandwidth
- **Distributing large FFTs across multiple GPU**
  - Communication is required when spreading and returning data
  - Significant amount collective communications
    - **Bandwidth and latency will be strong factors in determining performance**
- **Primary CUDA functions (used v9.1 for consistency across platforms)**
  - *cufftXtSetGPUs* **– identifies the GPUs to be used with the plan**
  - *cufftMakePlanMany64* **- Create a plan that also considers the number of GPUs available. The “64” means argument sizes and strides to be 64 bit integers to allow for very large transforms**
  - *cufftXtExecDescriptorC2C* **– executes C2C transforms for single precision**

Distribution A: This is approved for public release; distribution is unlimited

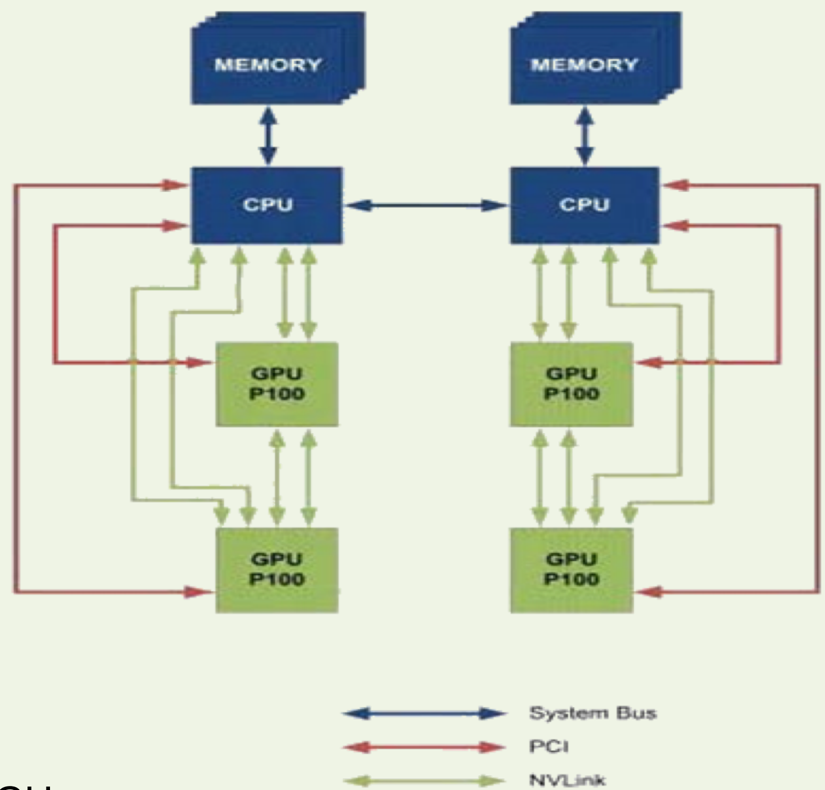
Hardware Configurations Examined

- **IBM Power 8**
  - Hokulea (MHPCC)
  - Ray (LLNL)
- **IBM Power 9**
  - Sierra (LLNL)
  - Summit (ORNL)
- **x86 PCIe**
- **Nvidia DGX-1 (Volta)**
- **Nvidia DGX-2**
- **Nvidia DGX-2H**



IBM POWER8 with P100 (Pascal) GPUs

- **2x P8 10 core processors**
- **4x NVIDIA P100 GPUs**
  - NVIDIA NVLink 1.0
    - 20 GB/s unidirectional
    - 40 GB/s bidirectional
  - 4 NVLink 1.0 lanes/GPU
    - 2 lanes between neighboring GPU
    - 2 lanes between neighboring CPU
- **X-Bus between CPUs**
  - 38.4 GB/s
- **POWER AI switch can be enabled**
  - Increases P100 clock speed from 1328 GHz to 1480 GHz

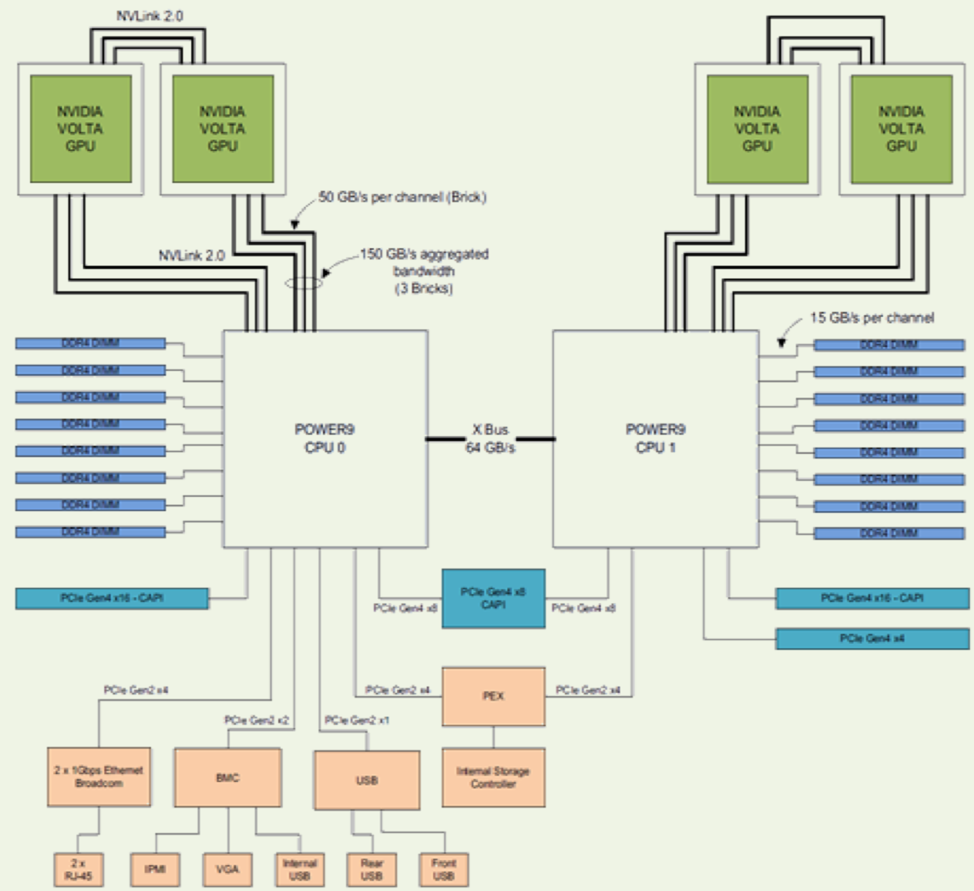


Distribution A: This is approved for public release; distribution is unlimited

Unclassified

IBM POWER9 with Volta GPUs

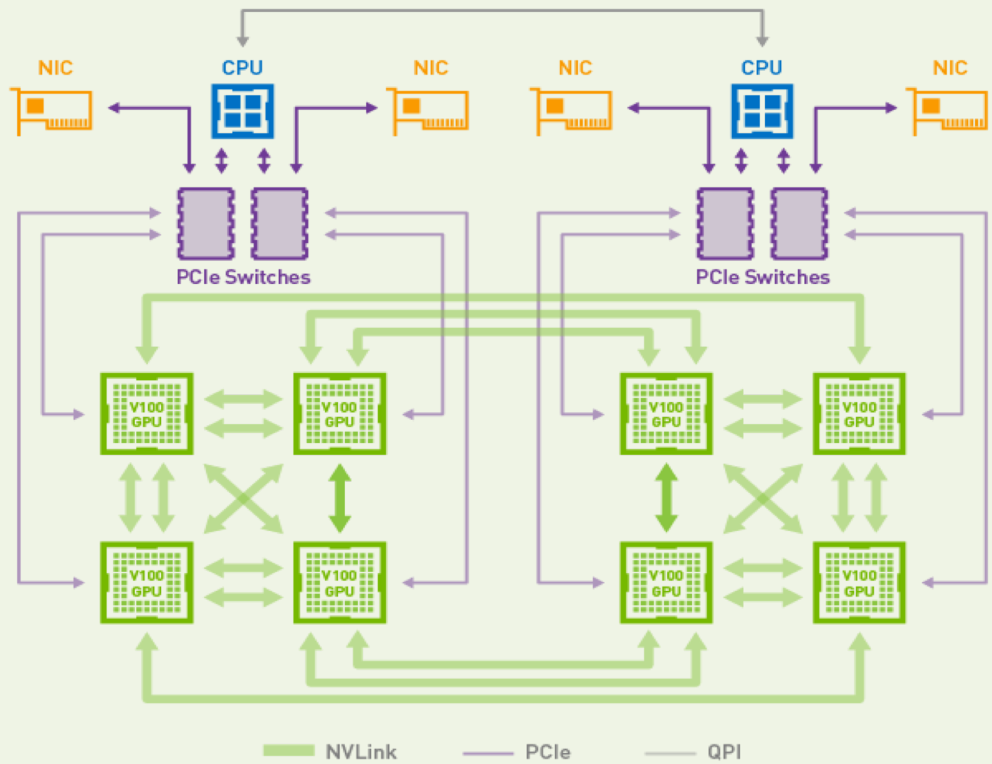
- **2x P9 22 core processors**
- **4x or 6x NVIDIA V100 GPUs**
  - NVIDIA NVLink 2.0
    - 25 GB/s unidirectional
    - 50 GB/s bidirectional
  - 6 NVLink 2.0 lanes/GPU
- **4x GPUs/node**
  - 3 lanes between neighboring GPU
  - 3 lanes between neighboring CPU
- **6x GPUs/node**
  - 2 lanes between neighboring GPU
  - 2 lanes between neighboring CPU
- **X-Bus between CPUs**
  - 64 GB/s



Distribution A: This is approved for public release; distribution is unlimited

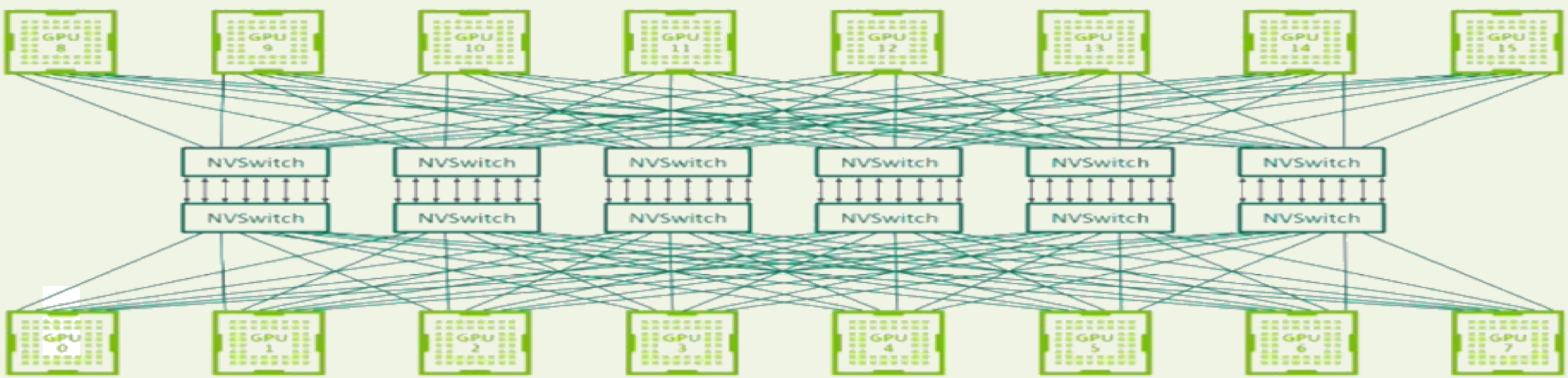
DGX-1v with 8 V100 GPUs

- **2x Intel Xeon E5-2698 v4, 20-core**
- **8x NVIDIA V100 GPUs**
  - NVIDIA NVLink 2.0
    - 25 GB/s unidirectional
    - 50 GB/s bidirectional
- **Hybrid cube mesh topology**
  - Variable lanes/hops between GPUs
    - 2 lanes between 2 neighboring GPUs
    - 1 lane between 1 GPU neighbor
    - 1 lane per cross CPU GPU
    - 2 hops to other cross CPU GPUs
  - PCIe Gen3 x16
    - 32 GB/s bidirectional
    - GPU & PCIe switch
    - PCIe switch & CPU



DGX-2 with 16 V-100s

- **2 Dual Intel Xeon Platinum 8168, 2.7 GHz, 24-cores**
- **16x NVIDIA 32GB V100 GPUs**
- **NVSwitch/NVLink 2.0 interconnection**
  - Capable of 2.4 TB/s of bandwidth between all GPUs
  - Full interconnectivity between all 16 GPUs



Distribution A: This is approved for public release; distribution is unlimited

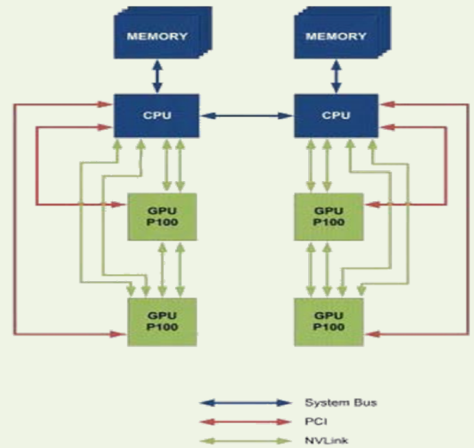
3D FFT (C2C) Performance Study

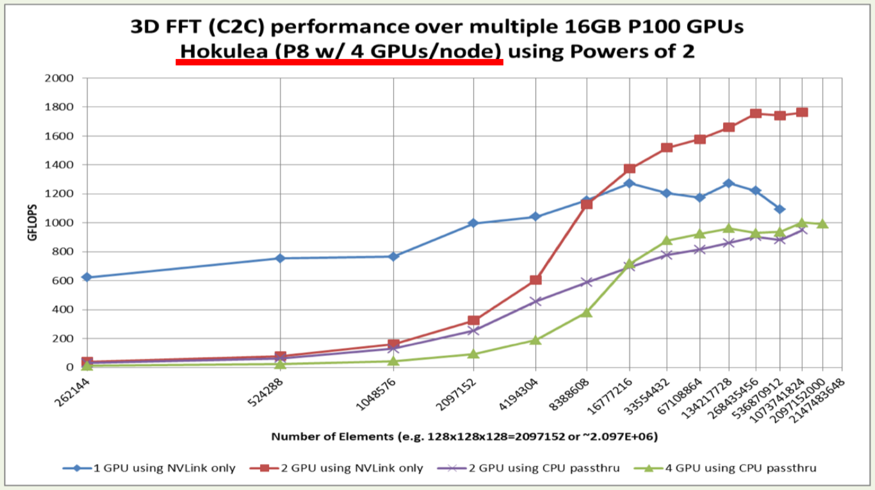
- **IBM Power Series**
  - IBM P8 (4x 16GB P100s) & IBM P9 (4x 16GB V100s)
  - Multiple sized cases from 64x64x64 to 1280x1280x1280 (memory limited)
  - 4 cases that shows how bandwidth & latency can affect performance:
    - 1 GPU only connect to CPU with NVLink
    - 2 GPUs attached to the same CPU and connected with NVLink
    - 2 GPUs attached to different CPUs
    - 4 GPUs (2 attached to each CPU)
- **x86 based systems**
  - Multiple sized cases from 64x64x64 to 2048x2048x2048 (memory limited)
  - PCIe connected GPU (no NVLink) system (PCIe G3 16x – 16GB/s bandwidth)
    - 1, 2, & 4 GPU cases
  - DGX-1v
    - 1, 2, 4, & 8 GPU cases
  - DGX-2
    - 1, 2, 4, 8, & 16 GPU cases

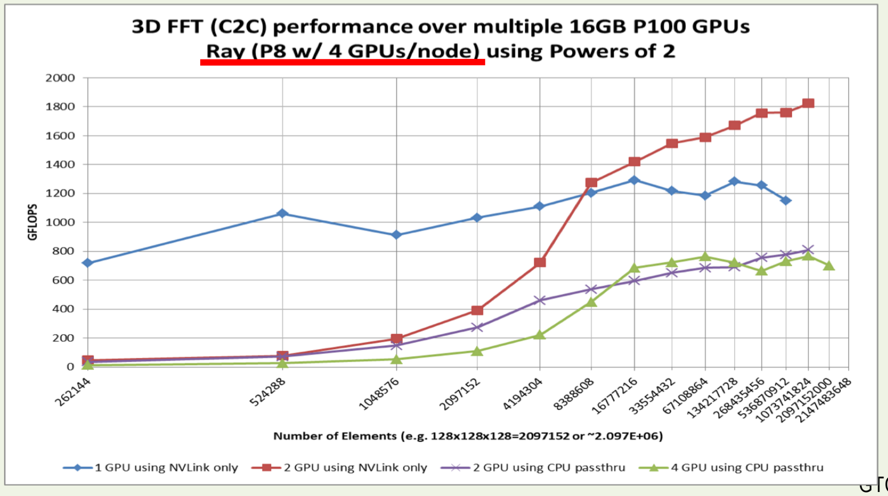
Distribution A: This is approved for public release; distribution is unlimited

IBM P8 Performance Study

- **Very similar performance between the 2 IBM P8s**
  - Only noticeable difference is the CPU pass-through cases
  - Better performance for non-CPU pass-through cases
  - Power AI: negligible effect as the limiting factor was bandwidth and latency
- **Same-socket 2x GPU case**
  - Bandwidth/latency has not dramatically affected performance before the problem size has reached its memory limit
- **All other GPU cases are more affected by bandwidth & latency**



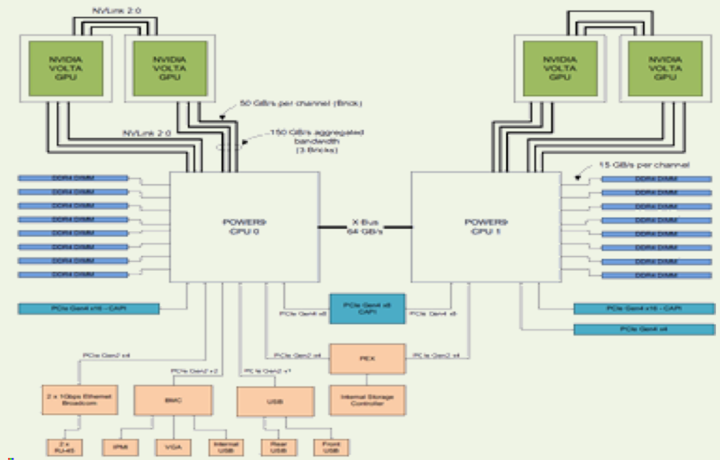


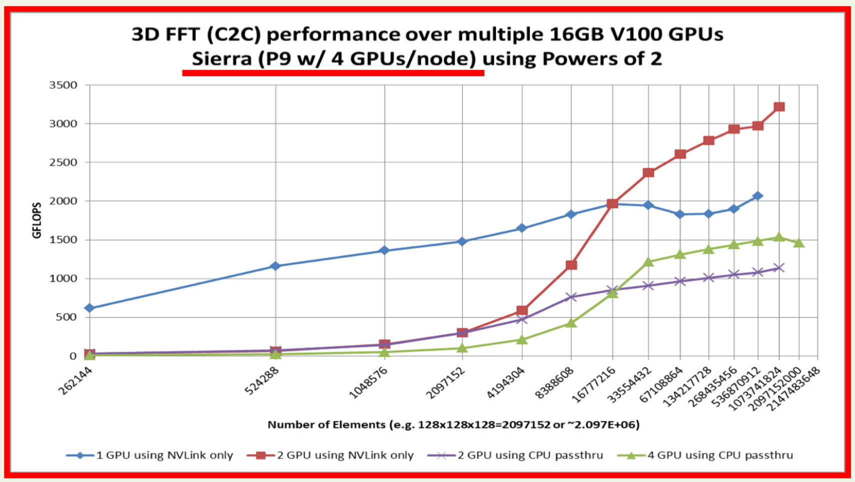


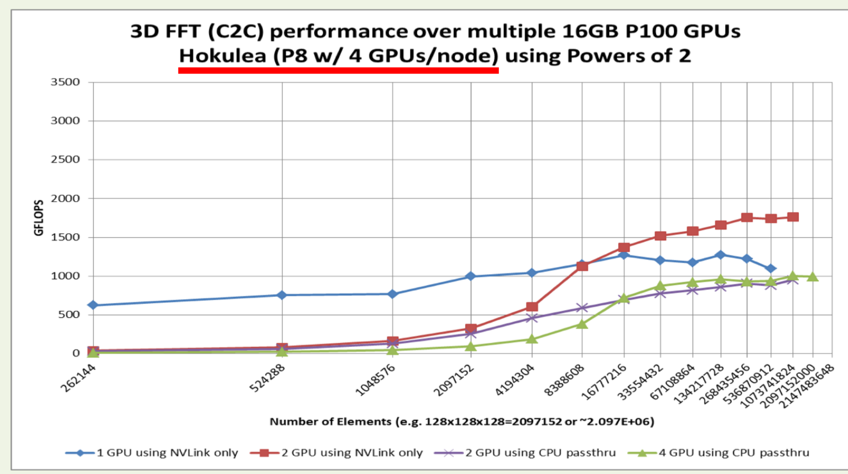
Distribution A: This is approved for public release; distribution is unlimited

IBM P9 Performance Study

- **P9 with 4x <u>16GB</u> V100s performed better than the P8**
  - Similar trends in performance as P8 b/c of architecture
  - Better overall performance b/c of V100 and 6 NVLink 2.0 lanes
  - Additional bandwidth of NVLink 2.0 allowed for better scaling
- **2x & 4x GPU CPU pass-through cases**
  - Bandwidth & latency limit performance gain
- **Summit performance expectation w/ 6 GPUs/node**
  - Less available lanes per GPU ≡ less bandwidth
  - Greater Memory (6x16GB) ≡ greater number of elements



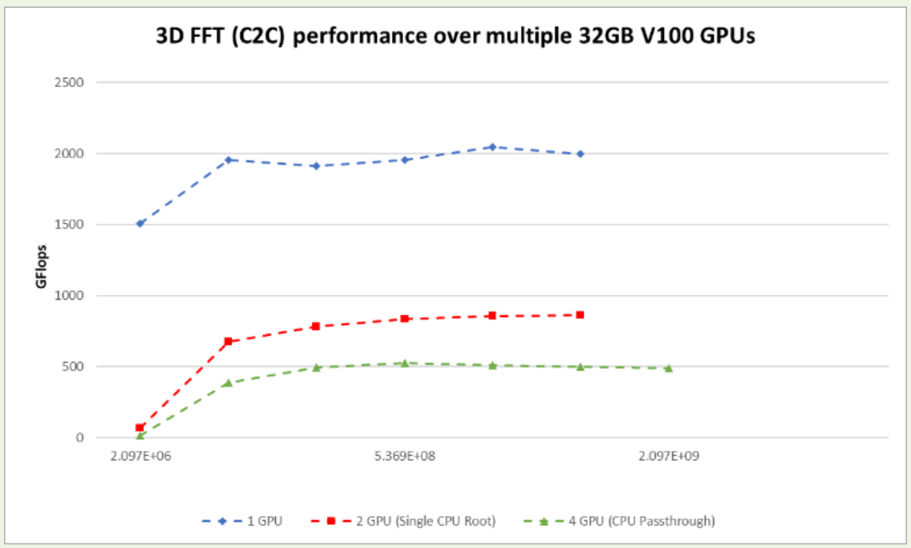


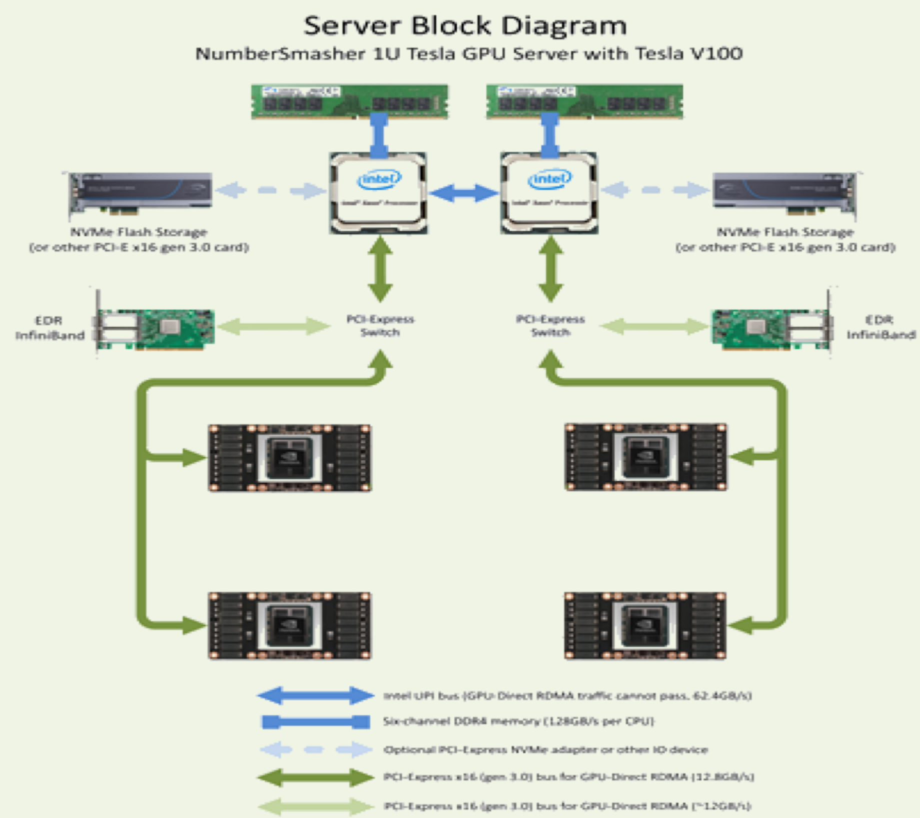


Distribution A: This is approved for public release; distribution is unlimited

x86 PCIe Based Performance Study

- **4x V100 (32GB) GPUs connected via x16 G3 PCIe (no NVLink)**
  – Communication saturates the PCIe bus resulting in performance loss
  – Also limited by the QPI/UPI communication bus
  – The 3D FFT does not scale

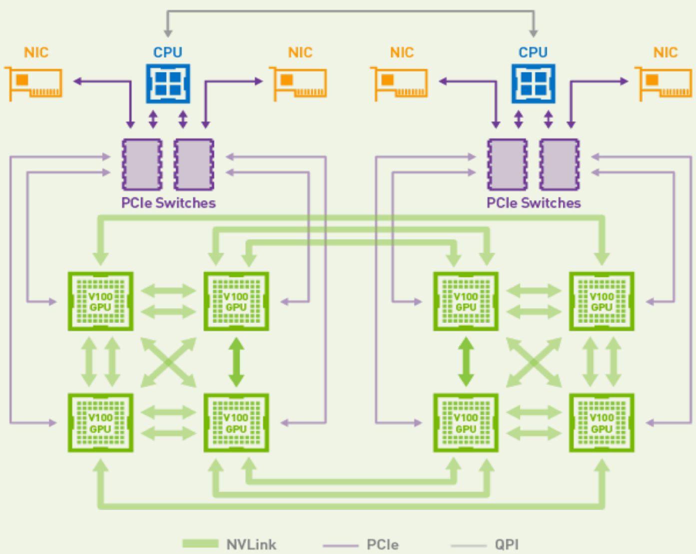


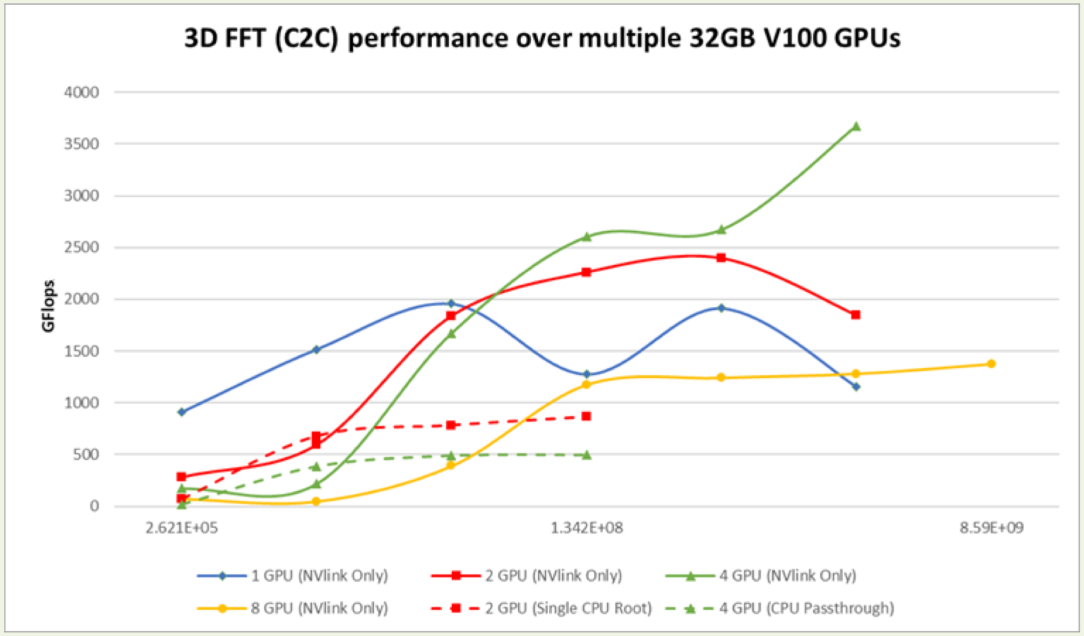


DGX-1v Performance Study

- **8x 32GB V100 GPUs**
  - 4 GPUs/CPU socket
- **Hybrid Mesh Cube topology**
  - Mix of NVLink connectivity

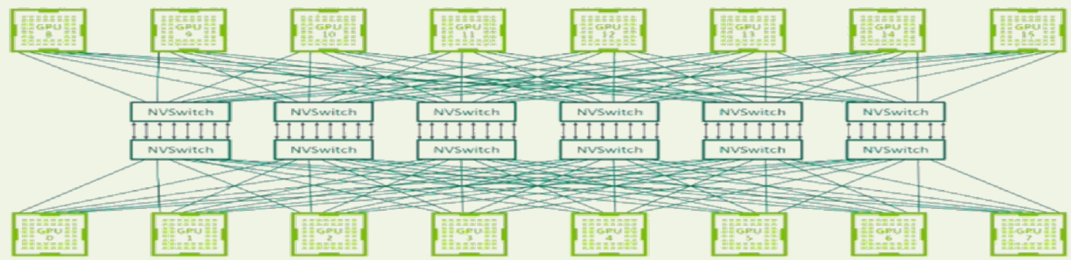
- **Variety of comm. cases**
  - NVLink 2.0
  - PCIe on same socket
  - PCIe with CPU pass-through

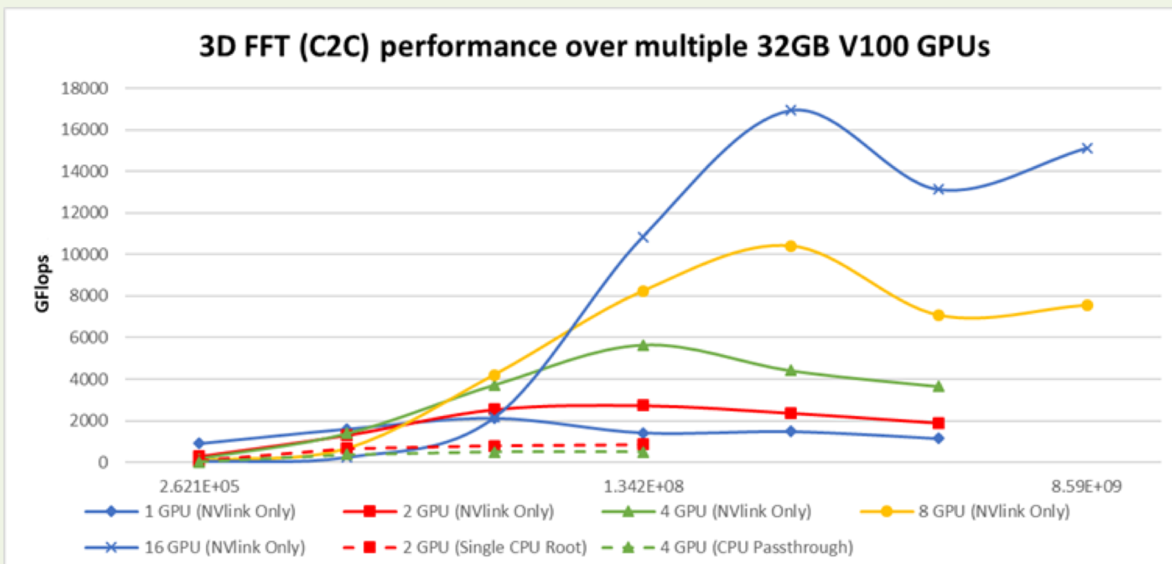




DGX-2 Performance Study

- **16x 32GB V100 GPUs**
  - NVSwitch/NVLink
- **Variety of comm. cases**
  - NVLink 2.0
  - PCIe on same socket
  - PCIe with CPU pass-through

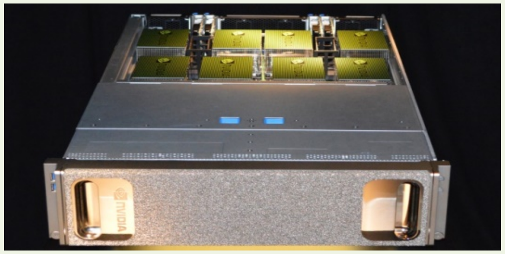


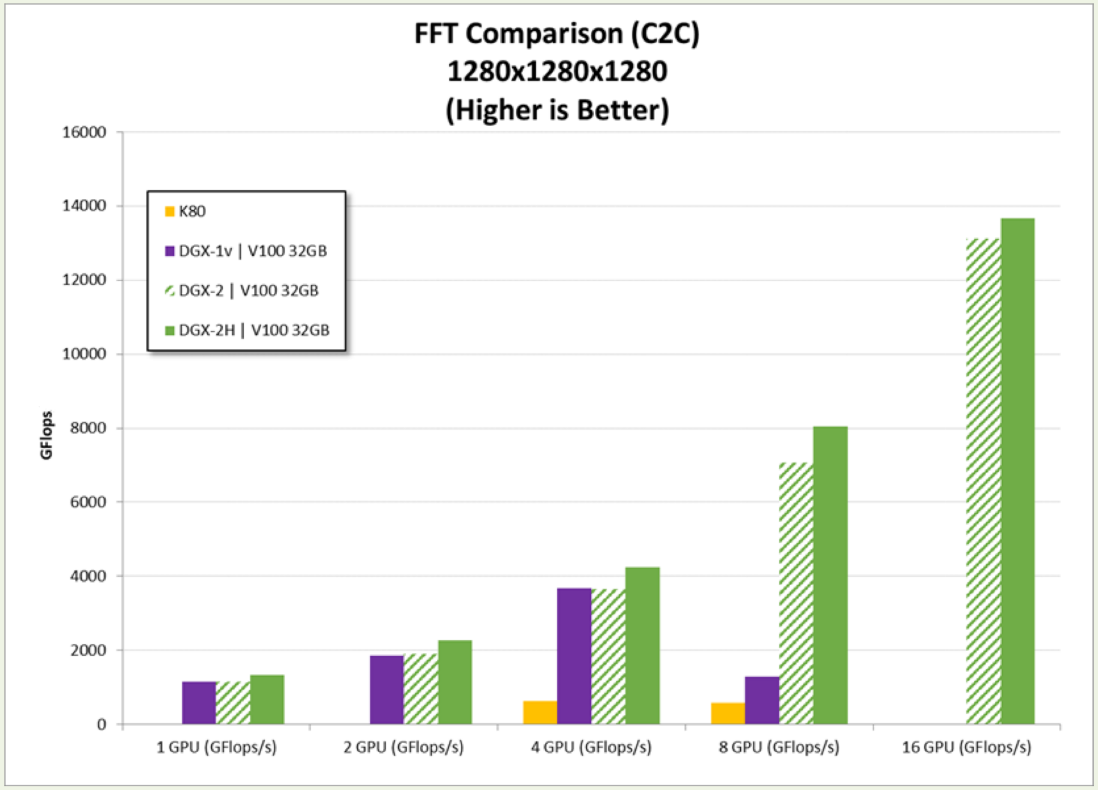


Distribution A: This is approved for public release; distribution is unlimited

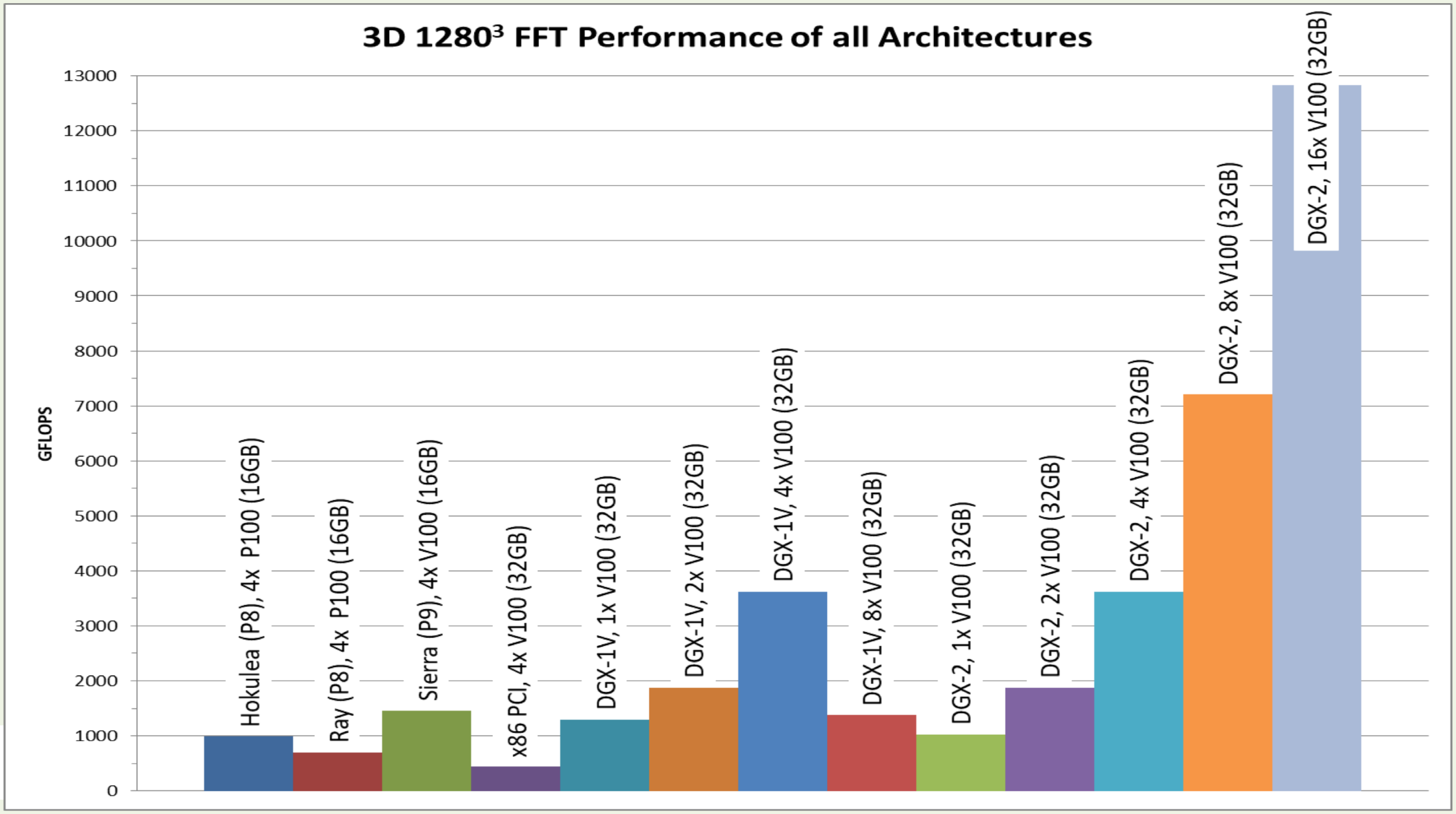
DGX-1v, DGX-2, DGX-2H Comparison

- **Key takeaways**
  - Very similar performance up to 4 GPUs
  - DGX-1v overhead for 8 GPU in the Hybrid Mesh Cube topology
  - DGX-2H performs ~10-15% better than the DGX-2



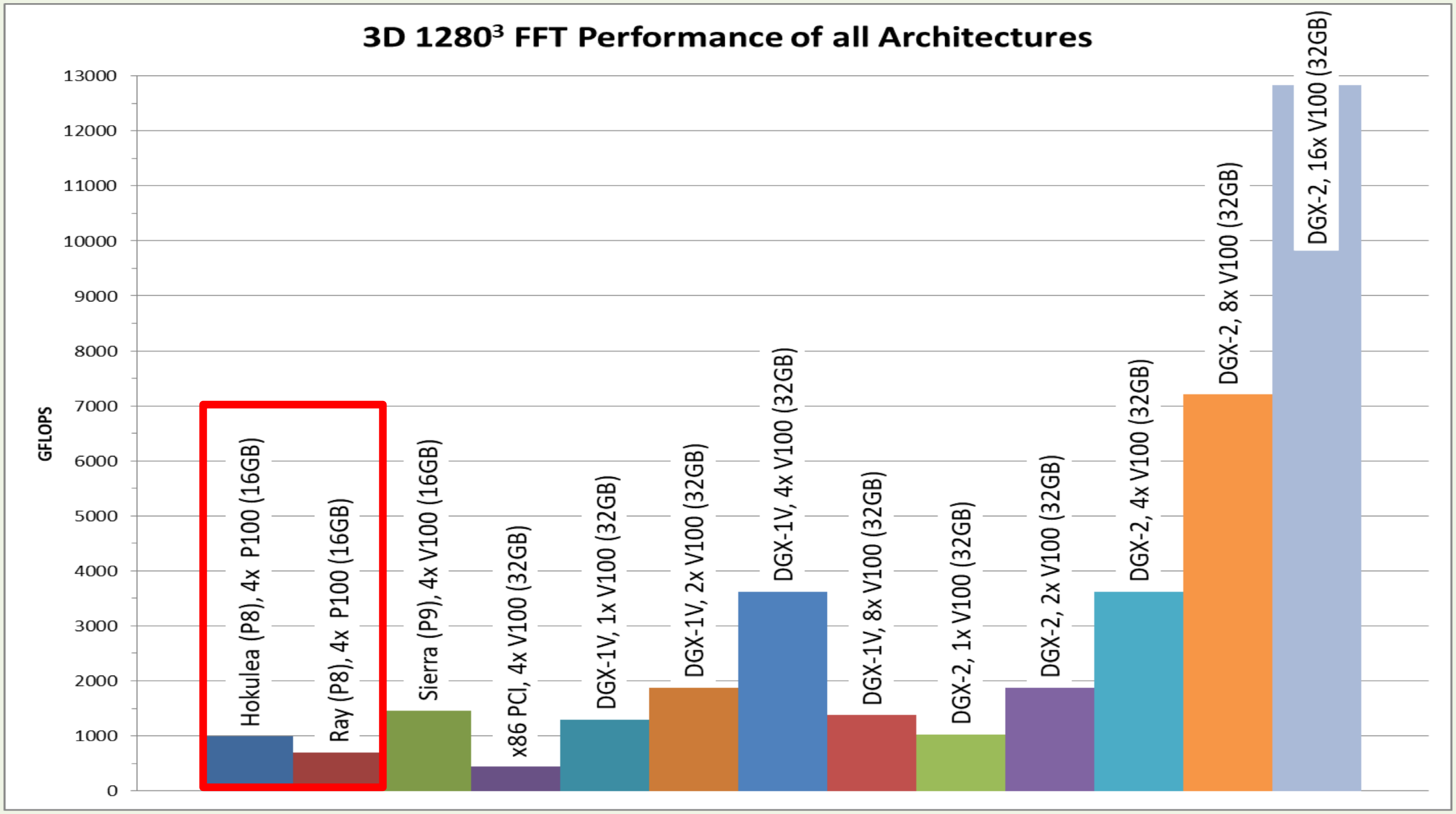


Collective Performance

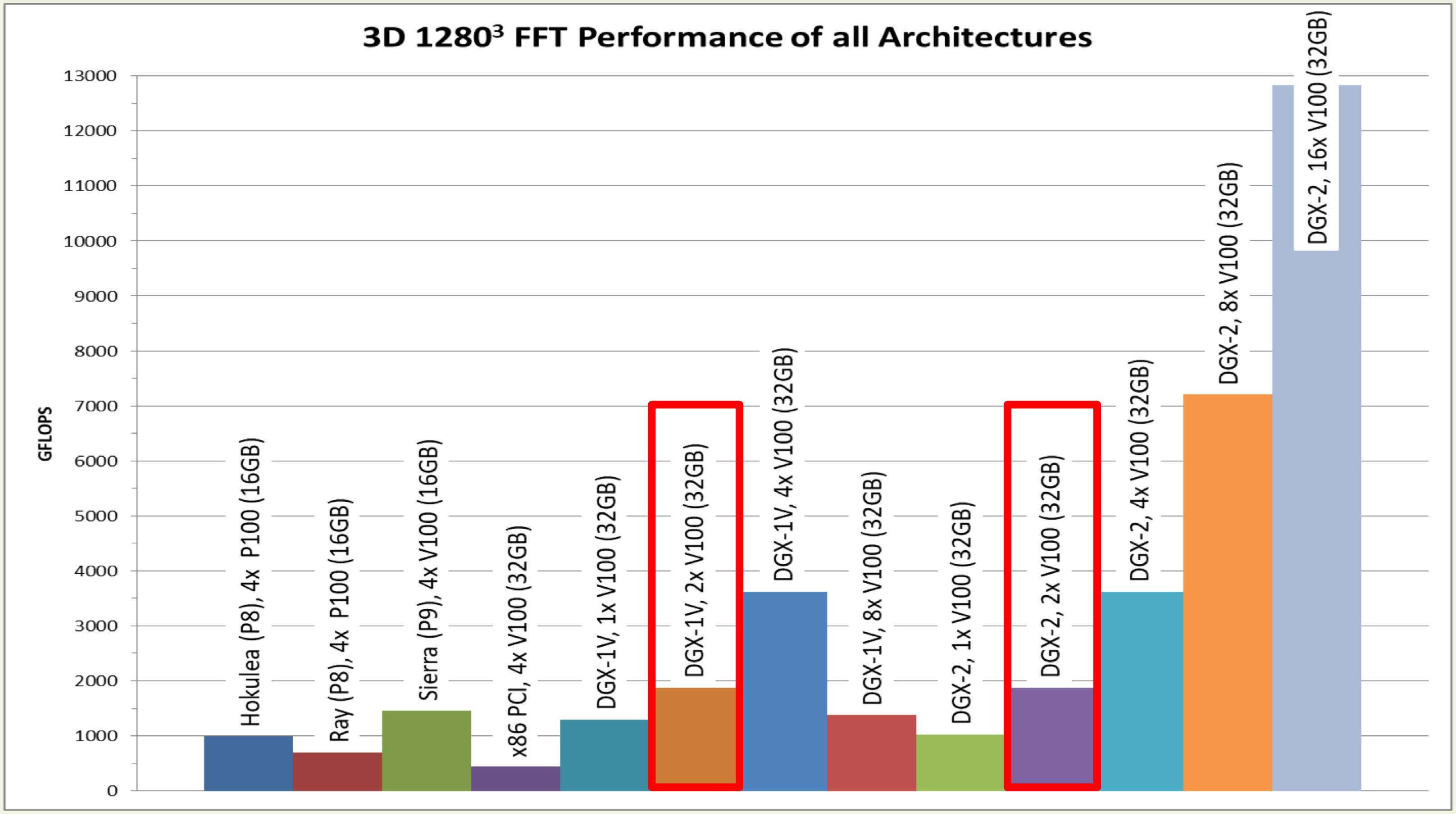


Distribution A: This is approved for public release; distribution is unlimited

4x P100s (16GB) Performance

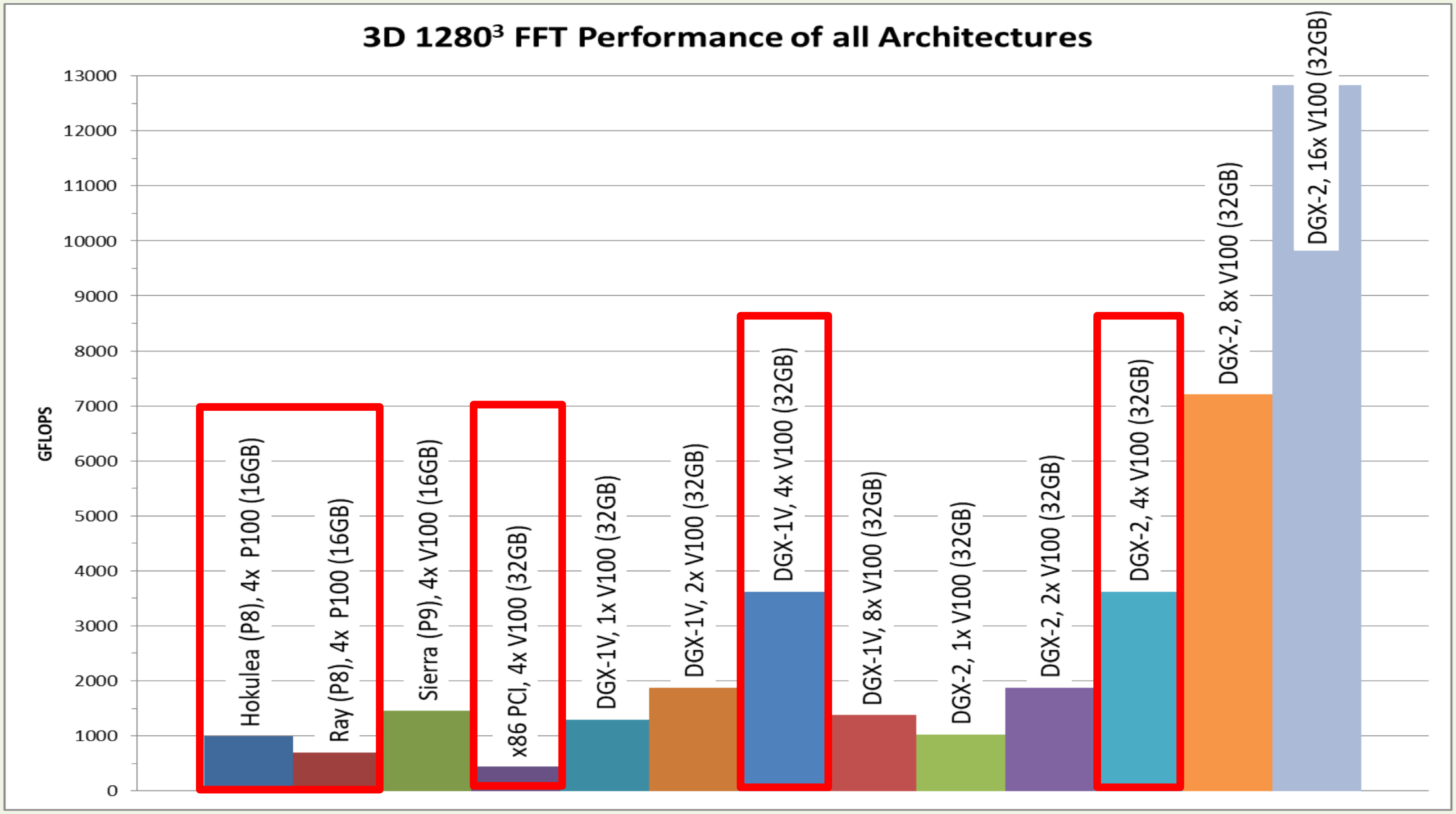


2x V100 (32GB) Performance



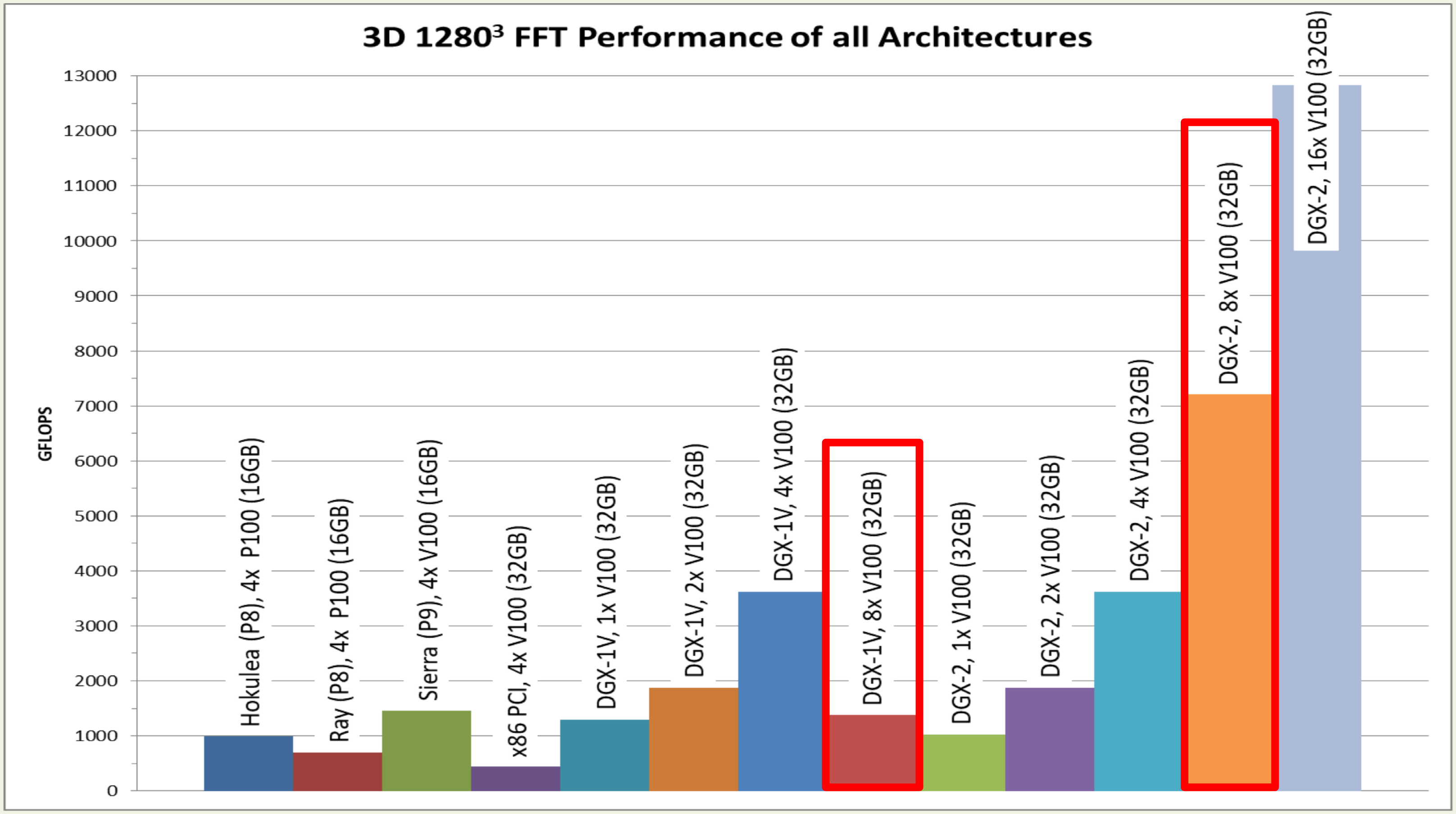
Distribution A: This is approved for public release; distribution is unlimited

4x V100 Performance



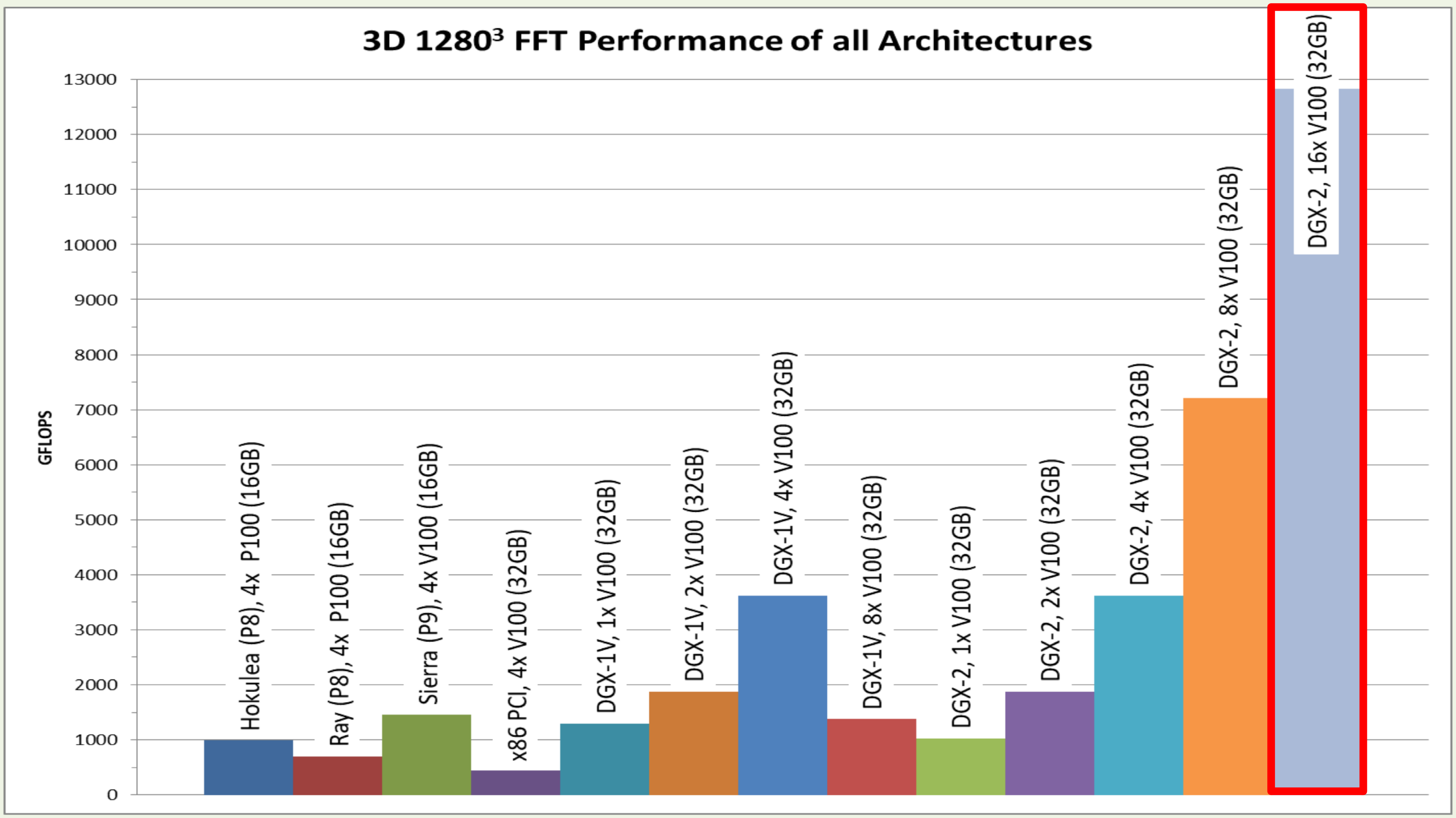
Distribution A: This is approved for public release; distribution is unlimited

8x V100 (32GB) Performance



Distribution A: This is approved for public release; distribution is unlimited

16x V100 (32GB) Performance



Distribution A: This is approved for public release; distribution is unlimited

Conclusions

- **Collective communication operations dominate performance when large FFTs are spread over multiple GPUs**
  - Highly dependent on underlying architecture's **bandwidth and latency**
- **x86 PCIe based systems**
  - Lower bandwidth and higher latency restrict scaling of multi-GPU FFTs
- **IBM Power Series**
  - Overhead associated when needed to handle communication between GPUs on different sockets limit performance
- **NVIDIA DGX-1v**
  - Hybrid Mesh Cube topology lowers communication overhead between GPUs
- **NVIDIA DGX-2**
  - NVSwitch technology has the lowest communication overhead between GPUs
- **NVIDIA DGX-2H**
  - Low communication overhead combined with faster GPUs

Distribution A: This is approved for public release; distribution is unlimited

Future Work

- **Future Work**
  - Examine Unified Memory FFT implementations
  - Multi-node Multi-GPU FFT implementations
  - Deeper analysis of the DGX-2H

- **Acknowledge support by**
  - The U.S. DoD High Performance Computing Modernization Program
  - The U.S. DoE at Lawrence Livermore National Laboratory
  - NVIDIA